In the Claims:

Please amend Claims 1, 2, 5, 6, 23, 24, 26, 27, 30, 31 and 33; and add new Claims 36-40, all as shown below. Applicants respectfully reserve the right to prosecute any originally presented or canceled claims in a continuing or future application.

Listing of Claims

1. (Currently Amended) A system for designing and executing processes, a business process, said system comprising:

an introspection module that generates a catalog of generic components by introspecting a set of exposed application programming interfaces (APIs) of a plurality of heterogeneous applications created in different programming languages and transforming a plurality of implementation-specific components of said heterogeneous applications into the generic components of said catalog wherein the catalog contains a series of entries created during introspection of the APIs, each entry representing a generic components component which, when invoked, are is bound to at least one of the implementation-specific components of said heterogeneous applications upon execution of the processes business process;

a component manager coupled to the introspection module and operable to manage said catalog generated by the introspection module by defining and organizing the generic components in said catalog; and

a process designer coupled to the component manager and operable to:

select at least one of the generic components from said catalog managed by the component manager; and

graphically construct a business process definition that includes a series of graphically represented activities linked by one or more transitions wherein at least one activity of said business process definition invokes the selected generic component from said catalog;

whereby a single process is assembled from existing components of multiple different programming languages, the components being

compiled modular routines of the heterogeneous applications that have been previously created in different programming languages; and

a repository for storing the ~~business~~ process definition; and

one or more process engines that execute said ~~business~~ process definition to instantiate a ~~business~~ process instance, wherein the ~~business~~ process instance interacts with the plurality of heterogeneous applications by invoking the generic components in said catalog and wherein the ~~business~~ process instance integrates the plurality of heterogeneous applications into ~~a~~ the single process by invoking services from the plurality of heterogeneous applications during execution of the activities of said process;

wherein during execution of the process instance, the multiple compiled routines of different heterogeneous applications are invoked in order to complete the activities of said process instance.

2. (Currently Amended)   The system of Claim 1, further comprising an organizational repository that includes said catalog, organizational data and a plurality of ~~business~~ processes generated by said process designer.

3. (Original)   The system of Claim 1, wherein the introspection module is operable to:

determine an implementation associated with at least one of the implementation-specific components;

retrieve the at least one of the implementation-specific components;

map each of the at least one of the implementation-specific components to a generic component to yield a mapping; and

save the mapping.

4. (Original)   The system of Claim 1, wherein the introspection module comprises a plurality of implementation modules, an implementation module operable to retrieve one or more implementation-specific components associated with an implementation.

5. (Currently Amended)   The system of Claim 1, further comprising a debugger coupled to the process designer and operable to detect an error of the ~~business~~ process.

6. (Currently Amended)   The system of Claim 1, further comprising:

a data warehouse coupled to the one or more process engines and operable to store transactional data describing the executed ~~business~~ process; and

a data server coupled to the data operable to organize the transactional data.

7-20. (Canceled)

21. (Previously presented)   The system of Claim 1 wherein said introspection module further includes at least one implementation module that is used to access implementation-specific components associated with at least one of: Java, Standard Query Language (SQL), Automation, Enterprise JavaBeans (EJB), CORBA, Remote Method Invocation (RMI), Extensible Markup Language (XML) schemas, Web Services and Java Naming and Directory Interface (JNDI).

22. (Previously presented)   The system of Claim 21 further comprising:

a binding table containing one or more entries that associate the selected implementation-specific components with generic components from said catalog.

23. (Currently Amended)   A computer implemented method for designing <u>and executing</u> ~~business~~ processes, said method comprising:

generating a catalog of generic components by introspecting a set of exposed application programming interfaces (APIs) of a plurality of heterogeneous applications implemented in multiple programming languages and translating implementation-specific components of the plurality of applications into generic components of said catalog wherein the catalog contains <u>a series of entries created during introspection of the APIs, each entry representing a</u> generic ~~components~~ <u>component</u> which, when invoked, ~~are~~ <u>is</u> bound to <u>at least one of</u> the

implementation-specific components of said <u>heterogeneous</u> applications upon execution of the <u>processes</u> ~~business process~~;

selecting at least one generic component from the catalog;

graphically constructing one or more ~~business~~ processes definitions, each ~~business~~ process definition including a series of graphically represented activities linked by one or more transitions wherein at least one activity of said ~~business~~ processes invokes the at least one generic component selected from said catalog<u>, whereby a single process is assembled from existing components of multiple different programming languages, the components being compiled modular routines of the heterogeneous applications that have been previously created in different programming languages</u>; and

executing the generated one or more ~~business~~ process definitions at one or more process engines in order to instantiate a ~~business~~ process instance, wherein the ~~business~~ process instance interacts with the plurality of applications by invoking the generic components of said catalog and wherein the ~~business~~ process instance integrates the plurality of heterogeneous applications into ~~a~~ <u>the</u> single process by invoking services from the plurality of heterogeneous applications during execution of the activities of said process.

24. (Currently Amended)   The method of Claim 23, further comprising:

providing an organizational repository that includes said catalog, organizational data and the ~~generated~~ one or more <u>process definitions</u> ~~business processes~~.

25. (Previously presented)   The method of Claim 23, further comprising:

determining an implementation associated with at least one of the implementation-specific components;

retrieving the at least one of the implementation-specific components;

associating each of the at least one of the implementation-specific components to a generic component to yield a mapping; and

saving the mapping.

26. (Currently Amended)   The method of Claim 23, further comprising:

detecting an error of the ~~generated~~ one or more process definitions ~~business processes~~ by

using a debugger.

27. (Currently Amended)   The method of Claim 23, further comprising:

storing transactional data describing the executed ~~business~~ process instance in a data

warehouse; and

organizing the transactional data at a data server.

28. (Previously presented)   The   method   of   Claim   23   wherein   translating

implementation-specific components of the plurality of applications into the generic components

of said catalog further includes:

associating a generic component identifier with an implementation-specific identifier and

storing the association in an entry of a binding table.

29. (Previously presented)   The method of Claim 23 wherein introspecting further

includes providing at least one implementation module that is used to access implementation-

specific components associated with at least one of: Java, Standard Query Language (SQL),

Automation, Enterprise JavaBeans (EJB), CORBA, Remote Method Invocation (RMI),

Extensible Markup Language (XML) schemas, Web Services and Java Naming and Directory

Interface (JNDI).

30. (Currently Amended)   A computer readable medium having instructions stored

thereon which when executed by one or more processors cause a system to:

generate a catalog of generic components by introspecting a set of exposed application

programming interfaces (APIs) of a plurality of heterogeneous applications

implemented in multiple programming languages and translating implementation-

specific components of the plurality of applications into generic components

of said catalog wherein the catalog contains <u>a series of entries created during</u> <u>introspection of the APIs, each entry representing a</u> generic ~~components~~ <u>component</u> which, when invoked, ~~are~~ <u>is</u> bound to <u>at least one of</u> the implementation-specific components of said <u>heterogeneous</u> applications upon execution of the <u>processes</u> ~~business process~~;

select at least one generic component from the catalog;

graphically constructing one or more ~~business~~ processes definitions, each ~~business~~ process definition including a series of graphically represented activities linked by one or more transitions wherein at least one activity of said ~~business~~ processes invokes the at least one generic component selected from said catalog<u>, whereby a</u> <u>single process is assembled from existing components of multiple different</u> <u>programming languages, the components being compiled modular routines of the</u> <u>heterogeneous applications that have been previously created in different</u> <u>programming languages;</u> and

execute the generated one or more ~~business~~ process definitions at one or more process engines in order to instantiate a ~~business~~ process instance, wherein the ~~business~~ process instance interacts with the plurality of applications by invoking the generic components of said catalog and wherein the ~~business~~ process instance integrates the plurality of heterogeneous applications into a single process by invoking services from the plurality of heterogeneous applications during execution of the activities of said process.

31. (Currently Amended)   The system of claim 1 wherein said ~~business~~ process definition is published to the repository before being deployed to the process engine.

32. (Previously presented)   The system of claim 1 wherein the catalog contains one or more entries, each entry including metadata that describes at least one of the plurality of implementation-specific components.

33. (Currently Amended)   The system of claim 1 wherein an activity of said ~~business~~ process definition connects to a subprocess that operates as a ~~business~~ process.

34. (Previously presented)   The system of claim 1 wherein said transitions indicate a next activity that is to be initiated after executing a previous activity.

35. (Previously presented)   The system of claim 1, further comprising:

a binding table that associates methods and attributes of the generic components in the catalog with the methods and attributes of the implementation-specific components of the applications.

36. (New)   The system of claim 1, further comprising:

a warehouse server that consolidates data from multiple process engines, wherein the warehouse server generates an N-dimensional cube to organize said data, wherein dimensions of the cube represent organizational data and measures of the cube represent transactional data of process instances.

37. (New)   The system of claim 37, wherein the measures of said cube are filtered by the dimensions of said cube in order to generate an analytical representation.

38. (New)   The system of claim 37 wherein the measures are identified with one or more of the following: a number of process instances in an activities, the number of instances completed by said process, execution time of a process instance and an average wait time for an activity.

39. (New)   The system of claim 37 wherein the dimensions are associated with one or more of the following: an organizational unit, a role, a user and time.

40. (New)   The method of claim 23, further comprising:

consolidating data from multiple process engines by a warehouse server, wherein the warehouse server generates an N-dimensional cube to organize said data, wherein dimensions of the cube represent organizational data and measures of the cube represent transactional data of process instances.